REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MAY 2013	CONFERENCE PAPER (Post Print)	DEC 2010 – NOV 2012

4. TITLE AND SUBTITLE

MEMRISTOR-BASED SYNAPSE DESIGN AND TRAINING SCHEME FOR NEUROMORPHIC COMPUTING ARCHITECTURE

5a. CONTRACT NUMBER
FA8750-11-2-0046

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)

Hui Wang, Hai Li, and Robinson Pino

5d. PROJECT NUMBER
T2NC

5e. TASK NUMBER
PO

5f. WORK UNIT NUMBER
LY

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Polytechnic Institute of NYU
2 MetroTech Center
Brooklyn, NY 11201

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/Information Directorate
Rome Research Site/RITB
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TP-2013-019

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved For Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Memristors have been rediscovered recently and then gained increasing attentions. Their unique properties, such as high density, nonvolatility, and recording historic behavior of current (or voltage) profile, have inspired the creation of memristor-based neuromorphic computing architecture. Rather than the existing crossbar-based neuron network designs, we focus on memristor-based synapse and the corresponding training circuit to mimic the real biological system. In this paper, first, the basic synapse design is presented. On top of it, we will discuss the training sharing scheme and explore design implication on multi-synapse neuron system. Energy saving method such as self-training is also investigated.

15. SUBJECT TERMS
Memristor; synapse; training

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			NATHAN MCDONALD
U	U	U	UU	6	19b. TELEPHONE NUMBER (*Include area code*) N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Memristor-based Synapse Design and Training Scheme for Neuromorphic Computing Architecture

Hui Wang and Hai (Helen) Li
Department of Electrical and Computer Engineering
Polytechnic Institute of New York University
Brooklyn, New York, USA
hwang07@students.poly.edu, hli@poly.edu

Robinson E. Pino
Air Force Research Laboratory, Information Directorate
Advanced Computing Architectures
Rome, New York, USA
robinson.pino@rl.af.mil

***Abstract*—Memristors have been rediscovered recently and then gained increasing attentions. Their unique properties, such as high density, nonvolatility, and recording historic behavior of current (or voltage) profile, have inspired the creation of memristor-based neuromorphic computing architecture. Rather than the existing crossbar-based neuron network designs, we focus on memristor-based synapse and the corresponding training circuit to mimic the real biological system. In this paper, first, the basic synapse design is presented. On top of it, we will discuss the training sharing scheme and explore design implication on multi-synapse neuron system. Energy saving method such as self-training is also investigated.**

Keywords – memristor; synapse; training

I. INTRODUCTION

The neuromorphic computing architecture that requires a large volume of memory and being adaptive to environment has demonstrated great potentials in developing high performance parallel computing systems [1]. Currently, most of research activities have been conducted at software or system level built upon the conventional Von Neumann computer architecture [2][3]. Developing the neuromorphic architecture at chip level by mimicking the biological system is another important direction. However, it results in high design complex and cost by using the traditional CMOS devices.

Though the existence of memristor was predicted in 1971 [4], the first physical realization was first reported thirty years later by HP Lab through a TiO_2 thin-film device[5]. Afterwards, more materials with memristive properties have been reported or rediscovered. The unique properties of memristor make it very promising to be used to mimic natural neuron networks [7]. First, the memristor-based memory can achieve an integration density as high as 10 Gbits/cm^2 [4][5]. Second, the memristor device has an intrinsic and remarkable feature called "pinched hysteresis loop", that is, the memristance relies on the total electric charge flowing through it [4][6]. Third, memristance remains unchanged when power is turned off.

Many memristor-based circuit designs have been explored, such as crossbar nonvolatile memory [8] and FPGA [9]. Strukov *et al.* integrated digital memory, programmable Boolean logic circuit and also neuron networks within a 3D hybrid CMOS/memristor structure. Rajendran *et al.* proposed memristor-based programmable threshold logic array [11] utilized it in synapse-neuron structure [12]. However, the training circuit and training scheme for memristor-based reconfigurable architecture design have not been fully explored yet.

In this paper, we proposed a single memristor-based synapse structure and the corresponding training circuit design. On top of it, we discussed the design optimization and its implementation in multi-synapse systems. With the aid of sharing training circuit and self-training mode, the performance and energy can be significantly improved. In the paper, we demonstrate the effective of the proposed synapse design by using TiO_2 memristor, though the design philosophy can be generalized to other memristor materials.

II. PRELIMINARY

In 1971, Professor Chua predicted the existence of the fourth fundamental circuit element, named as memristor, that uniquely builds the relationship between the magnetic flux (φ) and the electric charge (q) passing through the device as [4]:

$$d\varphi = M \cdot dq. \tag{1}$$

Considering that magnetic flux and the electric charge are integrals of voltage (V) and current (I) over time,

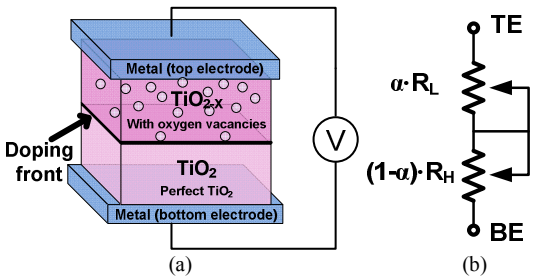


Figure 1: TiO_2 thin-film memristor. (a) Structure. (b) Equivalent circuit.

respectively. Eq. (1) can be expressed as

$$\begin{cases} V = M(\omega, I) \cdot I \\ d\omega/dt = f(\omega, I) \end{cases}, \quad (2)$$

where, $M(\omega, I)$ represents instantaneous memristance which varies over time and ω is a state variable.

Figure 1(a) illustrates the conceptual view of Pt/TiO_2/Pt structure: two metal wires (Pt) construct the top and bottom electrodes, and a thick titanium dioxide film is sandwiched in between. A perfect TiO_2 structure has a natural state as an insulator. However, the conductivity of oxygen-deficient titanium dioxide (TiO_{2-x}) is much higher. By moving the doping front under proper electrical excitations, the intermediate memristive state can be achieved. We use R_H and R_L to denote the total resistance when a TiO_2 memristor is fully undoped and doped, respectively. The overall memristance can be equivalent as two serially-connected resistances, as shown in Figure 1(b). That is

$$M(\alpha) = \alpha \cdot R_L + (1-\alpha) \cdot R_H . \quad (3)$$

Here, α ($0 \le \alpha \le 1$) is the relative doping front position, which is the ratio of doping front position over the total thickness of a TiO_2 device.

For demonstration purpose, we built our design based on TiO_2 thin-film memristor and adopted the device parameters from [5]. The memristance range is set from R_L=1KΩ to R_H=16KΩ. Note that the proposed can be applied to many memristor materials with different physical mechanisms.

III. The Principle of Memristor-Based Synapse

Rather than using memristor crossbar array in neuromorphic reconfigurable architecture, we propose a memristor-based synapse design to mimic the biological structure. Figure 2(a) depicts the conceptual scheme, which simply consists of a NMOS transistor (Q) and a memristor. When the input V_{in} is low, Q is turned off and the output V_{out} is connected to ground through the memristor. On the contrary, when V_{in} is high and turns on Q, memristance M and the equivalent resistance of Q (R_Q) together determine V_{out}:

$$V_{out} = f(V_{in} \cdot M). \quad (3)$$

Here, V_{out} is weighted by the memristance, which behaves like a synapse. Figure 2(b) shows the simulated V_{out} when sweeping the memristance from 1KΩ to 16KΩ. Here,

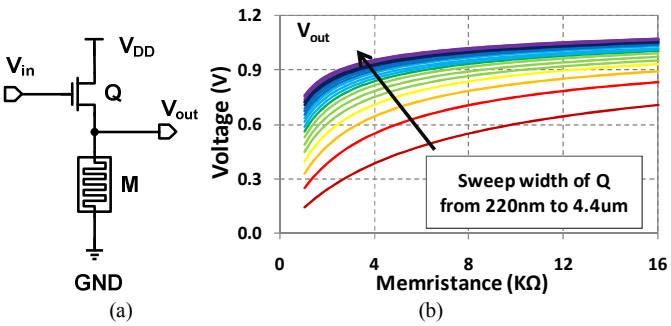
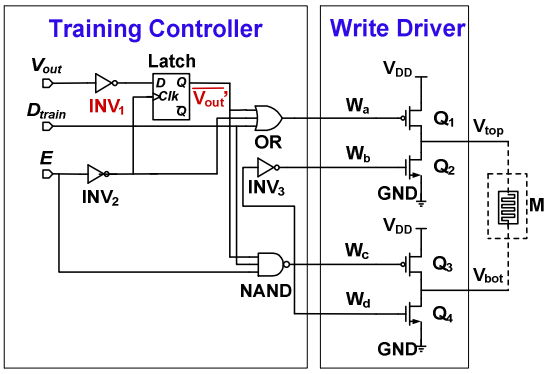
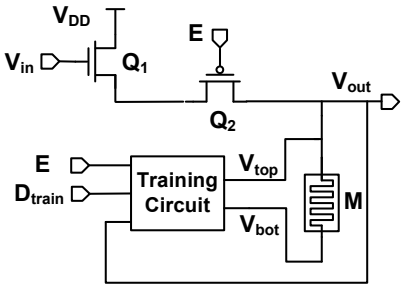


Figure 2: (a) Proposed synapse design. (b) Synapse output vs. memristance.



(a)



(b)

Figure 3: (a) The training circuit diagram. (b) The proposed synapse togteher with training circuit.

CMOS devices used TSMC 0.18μm technology.

Note that the response of the synapse design is dependent on the equivalent resistance of the transistor Q (R_Q), or, the size of Q. This can also be demonstrated in Figure 2(b) by sweeping the width of Q from 220nm to 4.4μm with a step of 220nm. The simulation shows that a larger Q can result in a wider range of V_{out} with poorer linearity. However, for a large Q, the enhancement of V_{out} by further increasing its size is marginal.

To improve design stability, a buffer can be added at output of the synapse to increase voltage swing. Furthermore, some circuit optimization techniques, such as asymmetry gate in other blocks, can be used to minimize the overall synapse-based system, as we shall show in Section IV.

IV. Synapse Training Circuit

A. Synapse Training Circuit Design

Being self-adaptive to the environment is one of the most important properties of a biological synapse. To accomplish the similar functionality, a training block is needed in the memristor-based synapse that can adjust its memristance.

1) Memristor Training Circuit

Figure 3(a) shows the diagram of training circuit for one synapse design, based on logic analysis and simplification. It includes two major components: *training controller* and *write driver*. By comparing the current synapse output V_{out} and the expected output D_{train}, training controller generates the control

TABLE I. TRAINING CIRCUIT OPERATION CONDITIONS.

E	V_{out}	D_{train}	V_{top}	V_{bot}	V_{mem}	Status
0	X	X	Floating	0	X	Operating
1	1/0	1/0	0	0	0V	No training
1	1	0	1	0	1.8V	R_H to R_L
1	0	1	0	1	-1.8V	R_L to R_H

* '0'– logic low; '1'– logic high, and 'X' – unknown or don't care.

TABLE II. SIZING OF INV_1 AND Q_1

P/N Ratio	PMOS/NMOS in INV_1	Q_1
2	720nm/360nm	18 × 220nm
	440nm/220nm	16 × 220nm
1	360nm/360nm	12 × 220nm
	220nm/220nm	11 × 220nm
0.5	360nm/360nm	9 × 220nm
	220nm/440nm	***9 × 220nm***

signals. The write driver uses these signals to control two pairs of NMOS and PMOS switches and supply training voltage pair V_{top} and V_{bot}. The training pair is applied to the two terminals of the memristor in the synapse design.

Determined by the training enable signal E, the training circuit can work under two modes.

- *Operating mode*: When $E=0$, the synapse is under regular operating (read) mode, and the training circuit is disabled.
- *Training mode*: The training circuit is enabled when $E=1$. By comparing the current synapse output V_{out} and the expected D_{train}, the training circuit generates V_{top} and V_{bot} applied to the two terminals of memristor to update or keep its memristance. We define $V_{mem}=V_{top}-V_{bot}$.

The training operation conditions are summarized in TABLE I.

Figure 3(b) depicts the proposed memristor-based synapse integrated with training circuit. An extra NMOS transistor Q_2 is inserted in synapse to isolate training operation from other voltage sources: when $E=1$, Q_2 is turned off so that the two terminals of memristor are controlled only by the training circuit, not affected by V_{in}.

The timing diagram of training circuit is demonstrated Figure 4(a). Before a training procedure starts, a sensing step is required to detect the current V_{out} to be compared with D_{train}. In the *sensing phase*, accordingly, training enable signal E is set to low for a very short period of time (*e.g., 4.5ns*) at the beginning of training. At the same time, $\overline{V_{out}}$ is sent to Latch, whose output $\overline{V_{out}}'$ remains constant during one training period, as shown in Figure 3(a). In the *training phase*, E is set back to high for a much longer time (*i.e.*, *51ms*) to change the memristance if needed.

We tested the training procedure by using the TiO_2 memristor model [5]. The training circuit was designed by using TSMC 0.18μm technology with V_{DD}=1.8V. Changing memristance from R_H to R_L or verse vice takes about *51ms*. The simulation result is shown in Figure 4(b). Here, the memristance is initialized as M=16KΩ. In the first *51ms*, it is trained to 1KΩ by setting D_{train} to low. Then at *t=51ms*, we set D_{train} to high and train the memristance back to R_H in the following *51ms*.

2) *Asymmetry Gate Design*

As we mentioned in section III, the size of Q_1 affects the range of V_{out}. Instead of adding buffer or having giant Q_1 in synapse, the asymmetry gate design can be adopted to minimize the layout area of synapse design. More specifically, we tuned P/N ratio of INV_1 in the training circuit (see Figure 3(a)). TABLE II summarizes the required sizes of INV_1 and Q_1 under different combinations that can make training successful. The result shows that the asymmetric design with P/N ratio =0.5 can obtain the smallest area. The last option is used in the following synapse analysis.

3) *Multi-synapse Training Scheme*

Most of the neuron systems are constructed by multiple synapses. In this section, we discuss the corresponding training scheme by taking a 2-synapse neuron in Figure 5 as the example. Here, A_1 and A_2 are two synapse inputs received from other neurons. M_1 and M_2 are memristor-based weights for two synapses S_1 and S_2. N is denoted for neuron with output V_{out}. The value of V_{out} depends on the functionality of N

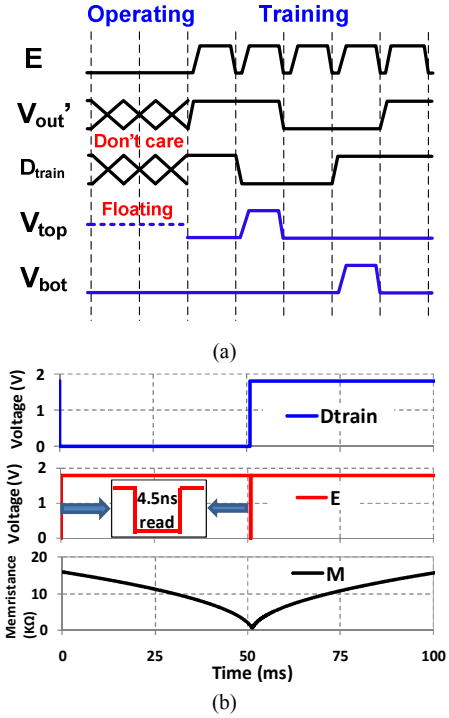


Figure 4: (a) The timing diagram of training circuit. (b) The simulation result of memristor training.

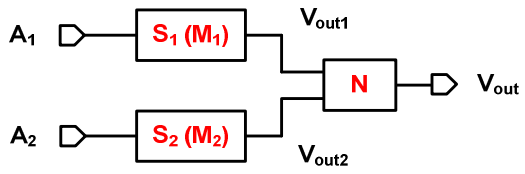


Figure 5: Two-input neuron structure.

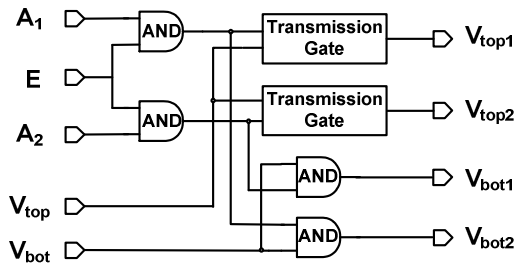


Figure 6:Training sharing distribution circuit.

as well as V_{out1} and V_{out2} from the two synapses. With the different combinations of M_1 and M_2, the two-input neuron could obtain different functionality.

To save design cost, memristances of the 2-synapse can be trained separately and share one training circuit. Figure 6 shows a training sharing distribution circuit, which generates training signals to control M_1 and M_2. The training sharing circuit operations under different conditions are shown in TABLE III.

The two synapse inputs A_1 and A_2 can be used to determine which memristor, M_1 or M_2, is in training. TABLE IV lists the required A_1 and A_2, when the logic functionality of N is one of the following: OR/NOR, XOR/XNOR, AND/NAND.

Compared to the separated training circuit for each memristor, the shared scheme can reduce 26% of training circuit transistor count. More saving in cost and area can be obtained when utilizing this training sharing distribution scheme to multi-synapse structure with more inputs.

4) Self-Training Mode

To improve training time and reduce power consumption, we introduce the concept of *self-training* in our design: rather than using a fixed long training period (*i.e.*, *51ms*), the self-training mode automatically stop programming memristor whenever V_{out} and D_{train} become same.

TABLE III. TRAINING SHARING CIRCUIT OPERATION

Status	V_{top1}	V_{bot1}	V_{top2}	V_{bot2}
Training M_1	V_{top}	V_{bot}	Floating	0
Training M_2	Floating	0	V_{top}	V_{bot}

TABLE IV. SYNAPSE INPUT PAIRS FOR DIFFERENT LOGICS

Functionality of N	Training M_1	Training M_2
OR/NOR	A_1=1, A_2=0	A_1=0, A_2=1
XOR/XNOR	A_1=1, A_2=0	A_1=0, A_2=1
AND/NAND	A_1=1, A_2=1	A_1=1, A_2=1

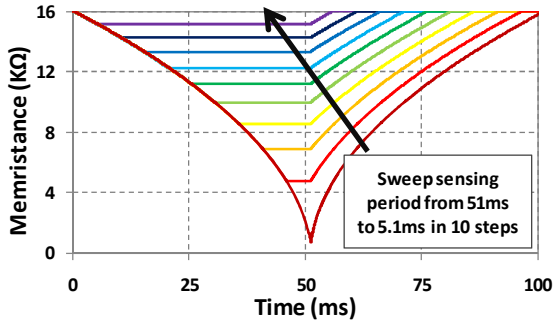


Figure 7: Self-training simulation.

TABLE V. SYNAPSE POWER CONSUMPTION ANALYSIS

Operation		Power	Energy
Read	R_L	1.04 mW	4.68 pJ
	R_H	113.4 uW	0.51 pJ
Training	From R_H to R_L	216.7 uW	11.1 uJ
	From R_L to R_H	234 uW	11.9 uJ

The proposed training circuit supports self-training mode by dividing a long training period into multiple shorter periods and detecting V_{out} in between. The programming period needs to be carefully selected: if it is too short, the delay and energy overheads induced by V_{out} detection may overwhelm the benefit of self-training. On the contrary, a long programming period cannot show enough benefit.

The simulation result in Figure 7 shows the memristance changing when sweeping programming period from *5.1ms* to *51ms* in 10 steps. Obviously, the self-training mode could significantly reduce training time. In theory, the proposed training circuit can train the memristance to any value between R_H and R_L. The real training time is determined by the specific application and neuron functionality.

5) Power Analysis

Reading and training simulation are conducted and power consumption data is collected in Table V. Energy is obtained when setting read time and write time as *4.5ns* and *51ms,* respectively.

V. CONCLUSION

Memristor has been proven as a promising device in neuromorphic architecture for its high-density, nonvolatility, and unique memristive characteristic. In this paper, we proposed a memristor-based synapse that can be used in neuromorphic computing architecture. The corresponding training operations including multi-synapse schemes and self-training have also been explored and discussed. The proposed synapse design can be generalized to other memristor materials for more applications. Next, we plan to further

explore and utilize the analog properties of the proposed synapse and develop the memristor synapse based neuron network.

ACKNOWLEDGMENTS AND DISCLAIMER

Contractor acknowledges Government support in the publication of this paper. This material is based upon work funded by AFRL under contract No. FA8750-11-2-0046. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL.

REFERENCES

[1] J. Partzsch, R. Schuffny, "Analyzing the Scaling of Connectivity in Neuromorphic Hardware and in Models of Neural Networks," *IEEE Transactions on Neuron Networks*, pp. 929-935, 2011.

[2] M. Wang, B. Yan, J. Hu, P. Li, "Simulation of large neuronal networks with biophysically accurate models on graphics processors," *International Conference on Neural Networks*, pp. 3184-3193, 2011.

[3] H. Shayani, P.J. Bentley, A.M.Tyrrell, "Hardware Implementation of a Bio-plausible Neuron Model for Evolution and Growth of Spiking Neural Networks on FPGA," *NASA/ESA Conference on Adeptive Hardware and Systems*, pp. 236-243, 22-25 Jun. 2008.

[4] L. Chua, "Memristor-the missing circuit element," *IEEE Trans. on Circuit Theory*, vol. 18, pp. 507-519, 1971.

[5] D. B. Strukov, et al., "The missing memristor found," *Nature*, vol. 453, pp. 80-83, 2008.

[6] X. Wang, et al., "Spintronic memristor through spin-torque-induced magnetization motion," *IEEE Electron Device Letters*, vol. 30, pp. 294-297, 2009.

[7] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale Memristor Device as Synapse in Neuromorphic Systems,"*Nano Letters*, vol. 10, no. 4, pp. 1297-1301, March 2010.

[8] Y. Ho, G. Huang, P. Li, "Nonvolatile memristor memory: Device characteristics and design implications," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 485-490, 2009.

[9] J. Cong, B. Xiao, "mrFPGA: A novel FPGA architecture with memristor-based reconfiguration," *IEEE/ACM Internaltional Symposium on Nanoscale Architectures (NANOARCH)*, pp. 1-8, 2011

[10] D. B. Strukov, et al., "Hybrid CMOS/Memristor Circuits," *IEEE International Symposium on Circuits and Systems (ISCAS),* pp. 1967-1970, 2010.

[11] J. Rajendran, H. Manem, R. Karri, and G. Rose, "Memristor based programmable threshold logic array," *IEEE/ACM International Symposium on Nanoscale Architecture*, pp. 5–10, Jun. 2010.

[12] G. Rose, R. Pino, Q. Wu, "Exploiting Memristance for Low-Energy Neuromorphic Computing Hardware," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2942-2945, 2011.